

## **REMARKS**

Claims 1-8, 13, 20-23, and 25-61 have been amended. Claims 1-61 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Objections to the Specification:**

The Examiner objected to the Title as being non-descriptive and suggested a new Title. The suggested Title includes “Within Virtual Machine Environment”; however, not all the independent claims recite a “virtual machine” as a limitation (see, e.g., claim 20). Applicants have provided an alternative replacement Title that is indicative of what is disclosed in the claims. Accordingly, removal of the objection to the Title is respectfully requested.

### **Section 112, Second Paragraph, Rejection:**

The Examiner rejected claims 1-61 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regard as the invention. Applicants respectfully traverse this rejection for at least the following reasons. First, Applicants assert that, contrary to the Examiner’s assertion, claims 1-61 clearly do “particularly point out and distinctly claim the subject matter which applicant regard as the invention.”

**In addition, the Examiner has not provided a proper *prima facie* § 112, second paragraph, rejection.** MPEP 2171, last paragraph, states (emphasis added):

If a rejection is based on 35 U.S.C. § 112, second paragraph, the examiner should further explain whether the rejection is based on indefiniteness or on the failure to claim what applicants regard as their invention. *Ex parte Ionescu*, 222 USPQ 537, 539 (Bd. App. 1984).

**The Examiner did not provide an explanation as to whether the rejection is based on indefiniteness or on the failure to claim what applicants regard as their**

**invention.** In fact, the Examiner provided no explanation or specific examples from the claims in regards to the § 112, second paragraph, rejection for any of the claims. Therefore, the § 112, second paragraph, rejection is improper and removal thereof is respectfully requested.

#### **Section 101 Rejection:**

The Examiner rejected claims 1-61 stood rejected under 35 U.S.C. § 101 as claiming an invention that is an abstract idea as defined in the case *In re Warmerdam*, 33 F3d. 1354, 31 USPQ2d 1754 (Fed. Cir. 1994). Applicants respectfully traverse this rejection for at least the following reasons.

In regards to claims 1-61, the Examiner “interprets that the claims 1-61 are non-statutory because they do not disclose that how a system will be able to carry out and execute its intended result without incorporating a processor, memory and medium...”.

In regard to claims 1-12, claim 1 has been amended to recite *a system comprising: one or more processors; and a memory comprising program instructions, wherein the program instructions are executable by the one or more processors to implement ....* Accordingly, removal of the § 101 rejection of claims 1-12 is respectfully requested.

In regard to claims 13-19, claim 13 has been amended to recite *a system, comprising: one or more processors; and a memory comprising program instructions, wherein the program instructions are executable by the one or more processors to implement ....* Accordingly, removal of the § 101 rejection of claims 13-19 is respectfully requested.

In regard to claims 20-26, claim 20 clearly recites *a system, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to implement...* Thus, contrary to the Examiner’s assertion including claims 20-26 in the § 101 rejection, claims 20-26 clearly

disclose “how a system will be able to carry out and execute its intended result” as the claims incorporate a processor and a memory. Therefore, for at least the reasons presented above, the § 101 rejection of claims 20-26 is improper and removal thereof is respectfully requested.

In regard to claim 27, the elements of claim 27 are all expressed as means for performing a specified function. Applicants remind the Examiner that under 35 U.S.C. § 112, paragraph 6:

An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of underlying structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

Thus, by statutory definition, the means claim specifically includes structure or material in support thereof and cannot be construed as software *per se*. Also, claim 27 has been amended to recite a computer system. Therefore, for at least the reasons presented above, the § 101 rejection of claim 27 is improper and removal thereof is respectfully requested.

In regard to claims 28-44, the claims have been amended to recite a computer-implemented method. Accordingly, removal of the § 101 rejection of claims 28-44 is respectfully requested.

In regard to claims 45-61, in addition to the above grounds of rejection, the Examiner further rejected the claims under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter since the claims recite “a computer-accessible medium.” Claims 45-61 have been amended to recite a “computer-accessible storage medium comprising program instructions, wherein the program instructions are computer-executable to implement...” Accordingly, removal of the § 101 rejections of claims 45-61 is respectfully requested.

### **Section 102(e) Rejection:**

The Examiner rejected claims 1-61 under 35 U.S.C. § 102(e) as being anticipated by Krapf et al (U.S. Patent 6,901,588) (hereinafter “Krapf”). Applicants respectfully traverse this rejection for at least the following reasons.

In regard to claim 1, Krapf discloses a system and method for sharing components between programming languages by use of polymorphic proxy (Krapf, Title). Krapf discloses a method and apparatus for representing and implementing a concept between two functional domains (e.g., programming languages) by using a proxy component in a first domain to wrap a [single] component of a second domain (Krapf, col. 2, lines 8-12). In contrast, claim 1 recites a proxy mechanism configured to generate, at runtime, a proxy to a particular one of different versions of an isomorphic interface provided by two or more subsystems. Krapf clearly does not anticipate what is recited in claim 1.

In further regard to claim 1, contrary to the Examiner’s assertion, Krapf does not disclose *a plurality of subsystems configured to execute within a virtual machine, wherein two or more of the plurality of subsystems provide different versions of an isomorphic interface to functions of the subsystems*. The Examiner cites Krapf, col. 39, lines 6-16 in support of this assertion. The cited paragraph discloses that a C++ proxy class may wrap a Java Listener interface for different subsystems. What Krapf is describing in the paragraph is a single “subsystem” (a Java Listener interface) that may be “wrapped” in a single C++ proxy class. What Krapf is describing is obviously quite different than *two or more of a plurality of subsystems that provide different versions of an isomorphic interface to functions of the subsystems*, as is recited in claim 1 of the instant application.

In further regard to claim 1, contrary to the Examiner’s assertion, Krapf does not disclose *a proxy mechanism configured to generate, for one of the plurality of subsystems at runtime of the one of the plurality of subsystems, a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems*. The Examiner

cites Krapf, col. 27, lines 46-48, in support of this assertion. The Examiner quotes “Assignment statements 258-266 achieve correct results through the same proxy mechanisms that cause assignment statement 256 to operate as the developer intended.” Applicants can find nothing in FIG. 11 or in the accompanying discussion thereof that includes the above-quoted selection that describes anything like what is recited in claim 1. Krapf does not teach or suggest a proxy mechanism configured to generate, for one of a plurality of subsystems at runtime of the one of the plurality of subsystems, a proxy to a version of an isomorphic interface provided by a particular one of two or more subsystems.

In further regard to claim 1, contrary to the Examiner’s assertion, Krapf does not disclose *wherein the proxy is configured to: receive a call to the isomorphic interface from the one of the plurality of subsystems; and convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems.* The Examiner cites Krapf, col. 42, lines 13-16, in support of this assertion. This selection is simply describing the overriding of a Java class call by a version of the call implemented in a “concrete” C++ class, and does not describe anything like what is recited in claim 1. Krapf is clearly not describing a proxy (generated at runtime by a proxy mechanism as recited in claim 1) that receives a call to an isomorphic interface from a subsystem and converts the call in accordance with a version of the isomorphic interface provided by a particular one of two or more subsystems.

In further regard to claim 1, contrary to the Examiner’s assertion, Krapf does not disclose *wherein the proxy is configured to: forward the converted call to the particular one of the two or more subsystems for execution.* The Examiner cites Krapf, col. 43, lines 49-61, in support of this assertion. The cited selection is describing an “error handling and exception handling” strategy that may be implemented in Krapf’s system. The cited selection does not describe anything like a proxy that is configured to forward a converted call to a particular one of two or more subsystems for execution.

Furthermore, col. 43, lines 49-61 of Krapf clearly describes a distinct and

different aspect of Krapf's system (an "error handling and exception handling" strategy) than is described in Krapf, col. 42, lines 13-16 (the overriding of a Java class call by a version of the call implemented in a "concrete" C++ class). Claim 1 recites a proxy (generated at runtime by a proxy mechanism as recited in claim 1) that is configured to: receive a call to the isomorphic interface from the one of the plurality of subsystems; convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems; and forward the converted call to the particular one of the two or more subsystems for execution. The Examiner is improperly relying on two different and distinct aspects of Krapf's system in asserting that Krapf discloses the above limitations as recited in claim 1. Neither of these citations teach or suggest what is recited in claim 1, nor do the two citations in combination teach or suggest what is recited in claim 1.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). For at least the reasons discussed above, Krapf clearly does not anticipate Applicants' claim 1.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 20, 27, 28, and 45.

**Regarding claims 2-6, 21-25, 29-32, 46-49, and 53, the Examiner has not provided a proper *prima facie* rejection of the claims.** These claims depend from independent claims 1, 20, 27, 28, and 45 respectively. The Examiner rejected these dependent claims along with the independent claims from which they depend. However, these dependent claims recite additional limitations not found in the independent claims, **limitations which the Examiner failed to address.** Therefore, the Examiner has not

provided a proper *prima facie* rejection of claims 2-6, 21-25, 29-32, 46-49, and 53.

**Regarding claims 13, 38, and 55, the Examiner has not provided a proper *prima facie* rejection of the claims.** These claims are independent claims. The Examiner rejected these claims along with and for the same reasons as independent claim 1. **However, these claims are of different scope and recite limitations not found in independent claim 1, limitations which the Examiner failed to address.** Therefore, the Examiner has not provided a proper *prima facie* rejection of claims 13, 38, and 55.

**Regarding claims 14-15, 18, 39, 41-43, 56, and 58-60, the Examiner has not provided a proper *prima facie* rejection of the claims.** These claims depend from independent claims 13, 38, and 55. The Examiner, however, rejected these claims along with and for the same reasons as independent claim 1. As noted above, the Examiner has not provided a proper *prima facie* rejection of independent claims 13, 38, and 55. Furthermore, these dependent claims recite additional limitations not found in the independent claims, **limitations which the Examiner failed to address.** Therefore, the Examiner has not provided a proper *prima facie* rejection of claims 14-15, 18, 39, 41-43, 56, and 58-60.

Regarding claim 7, contrary to the Examiner's assertion, Krapf does not disclose *wherein the proxy mechanism is further configured to provide an interface to the proxy mechanism for the plurality of subsystems, wherein the interface is accessible by the subsystems to specify isomorphic interfaces provided by other ones of the subsystems to be proxied by the proxy mechanism.* The Examiner cites Krapf, col. 8, lines 57-67, in support of this assertion. This selection from Krapf simply discloses that various interfaces (e.g. JNI, RNI, or JRI) may be used to "code the proxy layer of the C++ proxy component." Using one of several interfaces to code a proxy layer of a C++ proxy component is clearly and distinctly different than what is actually disclosed in claim 7 – a proxy mechanism providing an interface that is accessible by a plurality of subsystems to specify isomorphic interfaces provided by other subsystems. Krapf, in the cited selection or elsewhere, does not anticipate claim 7.

Thus, for at least the reasons presented above, the rejection of claim 7 is not supported by the cited prior art and removal thereof is respectfully requested.

Regarding claim 8, contrary to the Examiner's assertion, Krapf does not disclose *wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems using Java Reflection*. The Examiner cites Krapf, col. 15, lines 12-24, in support of this assertion. Applicants can find nothing in this citation, or elsewhere in Krapf, that refers to Java Reflection or the use thereof. Krapf, in the cited selection or elsewhere, does not anticipate claim 8.

Thus, for at least the reasons presented above, the rejection of claim 8 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 8 also apply to claims 17, 26, 33, 40, 50 and 57.

Regarding claim 10, contrary to the Examiner's assertion, Krapf does not disclose *wherein the one of the plurality of subsystems is an application, and wherein the two or more subsystems are versions of a runtime library*. The Examiner cites Krapf, col. 44, lines 38-49, in support of this assertion. This selection describes that "typical instances of Java Virtual machines reside in a shared library", that the shared library is "identified" and loaded, and then after the shared library is loaded, the JVM is started and initialized. Applicants note that the concept of a plurality of subsystems configured to execute within a virtual machine, wherein one of the subsystems configured to execute within the virtual machine is an application, and wherein two or more of the subsystems configured to execute within the virtual machine are versions of a runtime library that provide isomorphic interfaces is **clearly and distinctly different** than what Krapf discloses in the cited selection. Krapf, in the cited selection or elsewhere, does not anticipate claim 10.

Thus, for at least the reasons presented above, the rejection of claim 10 is not supported by the cited prior art and removal thereof is respectfully requested. Similar



remarks as those above regarding claim 10 also apply to claims 35 and 52.

Regarding claim 11, contrary to the Examiner's assertion, Krapf does not disclose *wherein the one of the plurality of subsystems and the two or more subsystems are applications*. The Examiner simply refers to the Krapf abstract in support of this assertion. Applicants can find nothing in the Abstract that teaches what is recited in claim 11. The Abstract simply states "Such proxy components may be used to gradually transform a digital entity (e.g., a software application) from a first digital domain to a second digital domain." Krapf is clearly not disclosing the concept of a plurality of subsystems configured to execute within a virtual machine, wherein one of the subsystems configured to execute within the virtual machine is an application, and wherein two or more of the subsystems configured to execute within the virtual machine are also applications that provide isomorphic interfaces.

Thus, for at least the reasons presented above, the rejection of claim 11 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 11 also apply to claims 36 and 61.

Regarding claim 12, contrary to the Examiner's assertion, Krapf does not disclose *wherein the plurality of subsystems are mobile agents*. The Examiner cites Krapf, col. 34, lines 55-67, in support of this assertion. Contrary to the Examiner's assertion, the cited selection teaches or suggests nothing about the concept of mobile agents as subsystems in a system. Applicants can find nothing in the cited selection, or elsewhere in Krapf that teaches what is recited in claim 12. Krapf is clearly not disclosing the concept of a plurality of subsystems configured to execute within a virtual machine, wherein one of the subsystems configured to execute within the virtual machine are mobile agents, and wherein two or more of the subsystems configured to execute within the virtual machine are also mobile agents that provide isomorphic interfaces.

Thus, for at least the reasons presented above, the rejection of claim 12 is not supported by the cited prior art and removal thereof is respectfully requested. Similar

remarks as those above regarding claim 12 also apply to claims 37 and 54.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-70000/RCK.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Applicant(s)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: October 2, 2007